Towards Realistic Benchmarking for Aircraft Engine Remaining Useful Life Prediction

Brendan Barnett[*1], Suhaas Nadella[*1], Truc Ho Nguyen[1], Ankith Nagabandi[1], Jai Goel[1], Jeremiah Lee[2], Danbing Seto[†2], Shihao Ji[1]

[1] *University of Connecticut,* [2] *Pratt & Whitney*

May 17, 2025

Abstract

Accurately estimating the Remaining Useful Life (RUL) of aircraft engines is critical for safety, maintenance optimization, and operational efficiency in aviation and other high-value industrial domains. While deep learning models have achieved strong performance on the NASA C-MAPSS benchmark, these simulated datasets fail to reflect the irregular sampling, sensor drift, and operational heterogeneity of real-world deployments. In this work, we present a large-scale evaluation of RUL prediction models on both C-MAPSS and a proprietary dataset from commercial airline operations. We release a curated version of the industrial dataset and benchmark five architectures – MLP, LSTM, LSTM with Additive Attention, Transformer, and CNN-LSTM – under identical preprocessing and training regimes. Our results reveal a sharp performance drop when transitioning from simulation to reality, underscoring the limitations of current benchmarks. While LSTM with additive attention achieves the best performance on C-MAPSS, it provides little advantage on sparse, noisy industrial data, where CNN-LSTM outperforms all models. These results highlight the importance of aligning model complexity with data quality. We further show that long-term degradation-aware feature engineering is essential for convergence in real-world settings. To support reproducibility and further research, all code and datasets are made publicly available at Engine-RUL.

1 Introduction

Predicting the *Remaining Useful Life* (RUL) of aircraft engines is critical to ensuring aviation safety, minimizing maintenance costs, and avoiding unplanned downtime. A single engine can exceed $10 million in acquisition cost, and each unplanned removal can ground an aircraft, ripple through global schedules, and expose operators to heavy losses and opportunity costs. Beyond aviation, robust RUL prediction forms the foundation of predictive maintenance strategies for any high-value, safety-critical asset. From power-generation turbines to offshore drilling equipment, this domain unlocks substantial economic and environmental benefits.

Research on RUL prediction has flourished around the NASA C-MAPSS simulation suite, which offers clean sensor streams, complete run-to-failure trajectories, and well-behaved degradation patterns. However, models that excel on C-MAPSS often falter when confronted with operational data marked by irregular sampling, sparse failures, mixed mission profiles, and sensor drift. Bridging the gap between academia and reality is essential for field deployment and real impact.

We close this benchmark–reality gap by pairing C-MAPSS with a large operational dataset[1] and conducting a comprehensive cross-validated study. Our core contributions are:

1. **Industry dataset.** We release a sanitized data set that captures real engine behavior and the challenges of sparse and noisy measurements, providing the community with a much needed test bed beyond toy simulations.

[*] Equal contribution. [†] The work was done while Dr. Seto was with Pratt & Whitney.

[1] The data were extracted from an unspecified fleet of engines operated under diverse conditions by multiple carriers in different regions; all records correspond to the same engine type.

2. **Rigorous 10-fold benchmark.** We evaluate five architectures – multilayer perceptron (MLP), vanilla LSTM, LSTM with additive attention, transformer, and temporal convolutional networks (CNN-LSTM) – under identical validation over C-MAPSS and 10-fold grouped cross-validation over the operational dataset, quantifying the performance cliff between synthetic and real conditions.

3. **Physics-guided feature engineering.** We introduce long-term degradation-aware features rooted in thermodynamic principles and catastrophe theory that extend the temporal context.

Results show that LSTM models augmented with additive attention deliver the most consistent performance across both benchmark and real-world datasets. Additionally, we find that encoding long-term degradation into fixed-length inputs through domain-informed features is essential for stable training and accurate prediction.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the C-MAPSS and industrial datasets alongside our preprocessing pipelines. Section 4 details the model architectures and baseline configurations. Section 5 outlines the experimental protocol used across datasets and the results. Section 6 offers a discussion of key findings, followed by conclusions with limitations and future directions in Section 7.

2 Related Work

2.1 RUL Estimation Methods

Remaining Useful Life (RUL) estimation has evolved through three primary methodological phases: statistical modeling, classical machine learning, and deep learning. Early statistical approaches modeled degradation with stochastic processes such as Wiener and Gamma models [1, 2], offering interpretability but limited adaptability to complex, nonlinear systems.

Classical machine learning introduced supervised algorithms like Support Vector Regression (SVR) [3], Random Forests [4], and ensemble methods [5], which improved performance on structured datasets but struggled with temporal dependencies in time-series sensor data.

Deep learning has since emerged as the dominant paradigm. Vanilla LSTMs, due to their ability to model long-term temporal dependencies, were widely adopted [6, 7]. CNN-based models restructured multivariate sensor data as spatial patterns, achieving strong performance on benchmark datasets [8, 9]. Recent architectures include dual-stage LSTMs [10], Bayesian recurrent networks [11], and Inception-based CNNs [12], each addressing different facets of uncertainty, multi-scale degradation, and complex operating conditions.

2.2 Attention Mechanisms in RUL Prediction

Attention mechanisms, originally developed for machine translation [13], have been successfully adapted to time series forecasting [14] and RUL prediction [15]. These mechanisms allow models to selectively weigh temporal features, improving interpretability and robustness against noise and variable-length sequences.

Additive attention has been used in multistage LSTM pipelines to highlight degradation-relevant timesteps [16], while more recent efforts explore sparse mixture-of-expert layers [17] and hybrid attention-CNN architectures [18] to handle data heterogeneity and missing values. This body of work affirms the efficacy of attention in isolating key failure indicators from multivariate signals, particularly in industrial settings.

2.3 Benchmarking on the C-MAPSS Dataset

The C-MAPSS dataset [19] remains a widely used benchmark for evaluating RUL algorithms. It simulates engine degradation under diverse operational profiles and fault modes, providing a controlled environment for algorithm comparison. There are four subsets to this dataset, FD001, FD002, FD003, and FD004. On FD001, Zheng et al. [10] achieved an RMSE of 16.14 using dual-stage LSTMs, later improved to 12.56 by Listou Ellefsen et al. [20] using semi-supervised learning strategies. Attention-based models such as Boujamza et al. [15] reported RMSEs of 13.95 on the same subset.

However, these results often fail to translate to real-world contexts. Unlike C-MAPSS, operational datasets feature incomplete sensor suites, irregular sampling, and highly variable degradation trajectories [21,

22]. This mismatch has sparked criticism that C-MAPSS overestimates generalization capability, motivating our benchmark on a proprietary industrial dataset. Our goal is to systematically evaluate how models tuned to C-MAPSS perform under realistic operating conditions with sparse, noisy, and non-stationary data.

3 Data and Preprocessing

3.1 NASA C-MAPSS

3.1.1 Overview

The NASA C-MAPSS dataset consists of four subsets (FD001-FD004) that vary in operating conditions and fault modes. Each subset contains multiple multivariate time series representing the run-to-failure data of different engine units. The data includes 21 sensor measurements and 3 operational settings recorded at regular intervals. FD001 and FD003 involve a single operating condition with one and two fault modes, respectively. FD002 and FD004 present six operating conditions with one and two fault modes, respectively. This progression in complexity allows for systematic evaluation of RUL prediction methods under increasingly challenging scenarios.

3.1.2 Preprocessing Pipeline

We applied a consistent, six-step pipeline to prepare both training and test data:

1. **Reproducibility.** Fixed the random seed for NumPy, TensorFlow, and Python's `random` module to ensure that data splitting, padding and model training are fully deterministic.

2. **RUL capping.** Clipped all training RUL values above 120 cycles to 120. This "kink" threshold prevents extreme remaining-life labels from skewing the loss and gradients.

3. **Training sequence generation.** For each engine unit, sorted its records by cycle and extracted overlapping windows of length 100. Each window retained only sensor features (dropping `unit_number` and RUL) and was labeled by the clipped RUL at its final time step.

4. **Test sequence construction.** For each test engine, we took the last 100 cycles. Engines with fewer than 100 records were front-padded with zero vectors so that all sequences match the length of 100. The true final RUL (uncapped) was used as the sequence label.

5. **Feature standardization.** Reshaped the 3D arrays of training windows into a 2D matrix and fit a `StandardScaler`. Applied the same transformation to both training and test feature windows, then reshaped back to $(N, 100, F)$.

6. **Target normalization.** Fitted a second `StandardScaler` on the training RUL labels (clipped) and applied it to normalize both training and test targets to zero mean and unit variance.

This procedure produces uniformly sized, standardized input sequences and scaled targets, ready for model training and evaluation.

3.2 Commercial Airline Dataset

3.2.1 Overview

The field-tested dataset obtained from commercial airline operations presents a substantial shift from traditional simulated benchmarks. This proprietary dataset comprises operational measurements from 98 engines across multiple aircraft types and airline fleets.

Unlike the C-MAPSS dataset, which is synthetically generated under controlled conditions, this dataset more accurately captures the complexities of real-world engine usage. It includes irregular sampling intervals, where measurements are recorded at varying frequencies depending on flight phase and operational demands. Sensor availability is inconsistent, with certain measurements missing entirely for some lifetimes or present

only during specific mission phases. The dataset also reflects a wide range of mission profiles, spanning both short-haul and long-haul operations with varying duty cycles. Sensor configurations and calibrations differ across aircraft, introducing additional variability.

While the dataset includes sensor modalities comparable to those in C-MAPSS, it differs significantly in the frequency and completeness of these measurements, offering a more challenging and realistic benchmark for RUL prediction.

3.2.2 Preprocessing Pipeline

To address the challenges of the industrial dataset and extract as much information as possible, we applied a structured pipeline to process dataset prior to training. The steps are outlined below:

1. **Engine identification.** Constructed a unique `engine_id` by concatenating the engine serial number (`esn`) and the usage event record ID (`UERID`). This identifier was used to group sequences by engine lifetime during training and validation.

2. **Feature engineering.** Computed two domain-informed cumulative degradation features: (1) `tot_dmg`, the total cumulative sum across 10 damage channels; (2) `tot_big_dmg`, the cumulative sum of the four most critical channels (`6,7,8,9`). By catastrophe theory, heavy damage indicators contribute to RUL in ways minimally correlated with light damage, so breaking them up is essential. Additionally, we introduced a normalized `timestamp` feature representing relative cycle position within each engine's lifetime.

3. **Metadata encoding.** Converted categorical metadata fields (encoded engine deployment information) into one-hot encoded vectors and appended them to the feature set. This allowed the model to distinguish different operating conditions and configurations across engines.

4. **Split data using complete engine lifetimes.** Applied 10-fold cross-validation using `GroupKFold`, ensuring that all data points from a given engine were kept within a single fold. This avoids data leakage and better reflects generalization to unseen engines.

5. **Sequence construction.** For each engine, created sliding windows of fixed length (`SEQ_LEN` = 100). Each sequence includes the most recent 100 time steps, front-padded with zeros if the engine history is shorter. The target label is the remaining useful life (RUL) at the final time step of each window.

6. **Normalization.** Standardized all numerical features using `StandardScaler`, fitting only on the training fold and applying the same transformation to validation data. The same procedure was applied to normalize RUL targets, ensuring consistent scale for model optimization.

This preprocessing strategy enabled consistent handling of heterogeneous engine lifetimes, metadata conditions, and degradation patterns within a realistic industrial setting.

4 Model & Baselines

4.1 Baselines

To benchmark our proposed approach, we implemented a suite of strong baseline models that span a range of architectural paradigms. Each baseline was trained and evaluated under consistent preprocessing and training regimes described in previous sections. `GridSearchCV` was utilized across all benchmarks to tune and optimize hyperparameters.

Multilayer Perceptron (MLP). The MLP baseline treats each input sequence as a flattened vector, discarding temporal ordering in favor of dense, fully connected representations. It comprises two hidden layers with 512 and 256 neurons respectively, both using `tanh` activations, batch normalization, and dropout for regularization. This model provides a baseline point of comparison for architectures that explicitly model temporal structure.

Transformer Encoder. This architecture projects the input sequence into a lower-dimensional embedding space and adds learned positional encodings. A single transformer encoder block – comprising

multi-head self-attention, residual connections, and a feed-forward network – processes the sequence. The output is aggregated using global average pooling, followed by a final prediction head. This model enables parallel sequence processing and flexible attention over time steps.

Vanilla Long Short-Term Memory (LSTM). This baseline uses a single unidirectional Long Short-Term Memory (LSTM) layer with 128 hidden units to process the input sequence and output a final hidden state, which is passed through a dense regression layer. The LSTM captures sequential dependencies in the data, providing a standard benchmark for temporal modeling.

Convolutional Neural Network LSTM (CNN-LSTM). The CNN-LSTM model first applies a stack of 1D convolutional layers to extract local temporal patterns within the input sequence, followed by an LSTM layer to capture longer-range dependencies. This hybrid architecture is designed to exploit both short-term trends and global temporal structure.

4.2 LSTM with Additive Attention (LSTM-A)

This section describes the LSTM with Additive Attention (LSTM-A) network used in our experiments and outlines the attention formulation, model configuration, and training protocol.

4.2.1 Additive Attention Formulation

Given an LSTM output sequence $H = [h_1, h_2, \ldots, h_T] \in \mathbb{R}^{T\times d}$, additive attention computes a context vector $c \in \mathbb{R}^d$ by first projecting each hidden state,

$$u_t = \tanh\big(W_h h_t + b_h\big), \qquad t = 1, \ldots, T, \tag{1}$$

then assigning a normalized weight

$$\alpha_t = \frac{\exp\big(v^\top u_t\big)}{\sum_{j=1}^{T} \exp\big(v^\top u_j\big)}, \tag{2}$$

and finally aggregating

$$c = \sum_{t=1}^{T} \alpha_t h_t. \tag{3}$$

Here $W_h \in \mathbb{R}^{d\times d}$, $b_h \in \mathbb{R}^d$, and $v \in \mathbb{R}^d$ are trainable parameters initialized with Glorot uniform initialization. The weights α_t highlight the most informative time steps for Remaining Useful Life (RUL) estimation.

4.2.2 Network Architecture

The final architecture processes a multivariate sequence $X \in \mathbb{R}^{T\times F}$ ($T = 100$):

1. **Input layer**: shape $(100, F)$.
2. **Unidirectional LSTM**: 128 hidden units, `return_sequences=True`, L2 weight decay 10^{-3}.
3. **Additive Attention**: custom Keras layer implementing (1)–(3), producing a fixed-length context vector.
4. **Dropout**: rate 0.2 for regularization.
5. **Dense output**: single linear unit for RUL prediction.

4.2.3 Implementation Details

The model is implemented in TensorFlow 2.7 with Keras:

- *Custom attention layer* built with `tf.tensordot` for efficient batched projections.
- *Optimizer*: Adam, learning rate 10^{-3}, gradient clipping `clipnorm`=1.0.
- *Loss*: Mean Squared Error (MSE).

4.2.4 Training Protocol

We train with batch size 100 for up to 100 epochs. Early stopping (patience 20) restores the best epoch, while `ReduceLROnPlateau` halves the learning rate if validation loss does not improve for 10 epochs (floor 10^{-5}). A 10-fold *GroupKFold* split, keyed on `engine_id`, prevents any engine's data from leaking across folds. Feature and target values are standardized within each fold. This setup allows the model to exploit long-term temporal structure while remaining robust to noise and variable operating conditions present in the industrial dataset.

5 Experimental Results

Our evaluation comprises two parts: (i) repeated train–test experiments on the four C-MAPSS subsets, and (ii) ten–fold cross-validation on the proprietary industrial dataset using group partitioning. All models were trained on a small GPU cluster; individual jobs were scheduled on a single GPU, so wall–clock times are comparable across runs. To support open-source research and facilitate reproducibility, we release all code and datasets at Engine-RUL.

5.1 C-MAPSS Dataset

C-MAPSS provides four independent train–test pairs (FD001–FD004). We strictly adhered to these predefined splits: for a given subset the model was trained on `train_FD00`i and evaluated on `test_FD00`i. To account for stochastic variation in initialisation and optimisation, every experiment was repeated ten times. The average RMSE and its standard deviation over these ten runs constitute the score for that subset. Results therefore reflect $4 \times 10 = 40$ independent C-MAPSS trainings for each model.

5.2 Industrial Dataset

Because the industrial data contain multiple records from the same physical engine, we adopted a *group* split to eliminate leakage. Each fold was created with `GroupKFold`(k=10), using the concatenated `esn+UERID` string as the grouping key. For every fold the model was trained on nine-tenths of the engines and evaluated on the remaining tenth. After completing all ten folds we reported the mean and standard deviation of the held-out RMSEs as the final score for each architecture.

All preprocessing, model hyper-parameters, and early-stopping criteria are held constant across folds and repetitions to ensure a fair comparison.

5.3 Results

Tables 1 and 2 present the RMSE scores (mean ± standard deviation) for each model across the NASA C-MAPSS and the proprietary industrial datasets.

Across all four NASA C-MAPSS sub-datasets, LSTM-A consistently outperforms competing models, achieving the lowest RMSE in every partition. Notably, it achieves a substantial margin on FD001 (14.47), where other models trail by 2–16 RMSE points. The performance gap is smaller in FD002, where MLP closely trails LSTM-A but suffers poor performance in other settings. Transformer-based models exhibit the highest instability, with large standard deviations – especially on FD001 – indicating sensitivity to initialization or overfitting. On average, LSTM-A leads with a 15% improvement over the 2nd-best architecture, validating its robustness and generalization across degradation modes.

On the industrial dataset, CNN-LSTM achieves the best performance with an RMSE of 532.30, outperforming all other models. Interestingly, LSTM and LSTM-A show nearly identical performance despite the latter's attention mechanism. This convergence may stem from the extreme sparsity and noise present in the industrial sensor data, which can limit the effectiveness of attention layers by reducing their ability to learn meaningful weighting across time steps. In such conditions, the inductive benefits of attention diminish, and simpler sequential models like LSTM suffice. Meanwhile, the Transformer and MLP models underperform, likely due to their inability to cope with irregular or incomplete time-series patterns without strong temporal priors.

Table 1: RMSE (± std) on NASA C-MAPSS Test Sets

C-MAPSS	LSTM-A	CNN-LSTM	LSTM	Transformer	MLP
FD001	**14.47 ± 0.70**	19.33 ± 1.99	16.40 ± 1.23	24.84 ± 7.13	30.46 ± 6.04
FD002	**30.52 ± 5.79**	32.86 ± 2.75	35.65 ± 3.67	35.12 ± 4.55	30.95 ± 0.72
FD003	**15.12 ± 1.00**	18.41 ± 1.54	17.26 ± 1.38	18.69 ± 4.36	24.99 ± 3.27
FD004	**27.18 ± 1.50**	33.24 ± 1.44	33.32 ± 3.46	37.55 ± 1.11	46.54 ± 0.93
Avg RMSE	**21.82 ± 2.25**	25.96 ± 1.93	25.66 ± 2.44	29.05 ± 4.29	33.24 ± 2.74

Table 2: RMSE (± std) on the Industrial Dataset

Dataset	LSTM-A	CNN-LSTM	LSTM	Transformer	MLP
Industry	600.11 ± 117.17	**532.30 ± 132.90**	625.94 ± 153.01	681.28 ± 140.52	652.06 ± 176.20

6 Discussion

Our findings underscore several critical insights for the future of RUL prediction and the broader prognostics community.

The Benchmark Gap. There is a substantial and persistent gap between performance on synthetic benchmarks like C-MAPSS and real-world industrial data. Models that achieve low error on C-MAPSS often degrade significantly when exposed to operational variability, noise, and incomplete signals in commercial engine environments. This highlights the urgent need for more realistic benchmarks. Our work addresses this by introducing a challenging new testbed grounded in real airline data – one that better reflects the complexity of actual deployment settings.

Effectiveness of Attention Mechanisms. LSTM with Additive Attention achieved the most consistent and robust performance on the C-MAPSS benchmark, outperforming all other models across all four sub-datasets. Its ability to dynamically weight time steps allows it to highlight key degradation patterns while minimizing the influence of irrelevant signals – a clear strength in rich, structured data. However, on the sparse and noisy industrial dataset, LSTM-A offered no meaningful improvement over the baseline LSTM, suggesting that its attention mechanism may be less effective when temporal signals are irregular or degraded. These results indicate that attention enhances performance under clean and informative conditions but may offer limited benefit in settings with high data sparsity.

Role of Feature Engineering. Despite the promise of end-to-end learning, we found that domain-specific feature engineering was indispensable. Without features that encode cumulative effects and operational context, such as cumulative damage and mission-normalized timestamps, no model was able to converge reliably. This is particularly important given that each model only observes a short temporal context window; engineered features serve to inject long-term degradation signals into that limited view. In effect, they summarize prior operational history that would otherwise be inaccessible, enabling models to make meaningful RUL predictions despite truncated inputs.

Toward Realistic Evaluation and Deployment. RUL estimation is a high-impact problem with direct implications for safety, maintenance planning, and cost optimization in aviation and beyond. We hope this new benchmark accelerates progress by providing a far more realistic setting for model development and evaluation. By closing the gap between research environments and real-world conditions, we aim to foster models that are not only accurate in theory, but deployable in practice.

7 Conclusion

This work presents a new benchmark for Remaining Useful Life (RUL) prediction using real-world engine data from commercial aviation, addressing the gap between synthetic benchmarks like NASA's C-MAPSS and the challenges of operational environments. Our findings highlight a stark performance drop when moving from structured, well-behaved datasets to noisy, sparse, and irregular industrial data. To facilitate

realistic evaluation, we release a high-variance testbed that better reflects practical degradation patterns and sensor irregularities encountered in the field.

Among the evaluated models, LSTM with Additive Attention consistently outperformed others on the C-MAPSS benchmark, leveraging its ability to prioritize critical time steps in rich, structured sequences. However, on the industrial dataset, LSTM-A showed no significant improvement over standard LSTM, suggesting that attention mechanisms may lose effectiveness in highly sparse and noisy conditions. These results underscore the importance of aligning modeling strategies with data characteristics and reaffirm the value of domain-informed feature engineering, which was essential for model convergence and generalization.

Limitations and Future Work While our benchmark captures important aspects of real-world RUL prediction, it is limited in scale and scope relative to global fleets. The need for manual feature extraction also restricts applicability across different machinery types. Additionally, we did not explore recent hybrid or self-supervised models that could better adapt to low-resource settings. As for future work, to enhance generalization, we plan to investigate data augmentation strategies and pretraining across diverse fleets. Incorporating contextual signals such as maintenance logs and environmental factors will enable richer modeling and support advanced architectures like multimodal encoders. By continuing to close the gap between benchmarks and reality, we hope this benchmark supports the development of RUL models that are not just accurate, but deployable.

References

[1] C. Valdez-Flores and R. M. Feldman. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics*, 36(4):419–446, 1989.

[2] E. Zio and F. Di Maio. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, 95(1):49–57, 2010.

[3] C. Ordóñez, F. S. Lasheras, J. Roca-Pardiñas, and F. J. de Cos Juez. A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 346:184–191, 2019.

[4] J. Z. Sikorska, M. Hodkiewicz, and L. Ma. Prognostic modeling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836, 2011.

[5] C. Zhang, P. Lim, A. Qin, and K. C. Tan. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2306–2318, 2017.

[6] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 275:167–179, 2018.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[8] X. Li, Q. Ding, and J. Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018.

[9] G. S. Babu, P. Zhao, and X.-L. Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. In *Proc. International Conference on Database Systems for Advanced Applications*, pages 214–228. Springer, 2016.

[10] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta. Long Short-Term Memory Network for Remaining Useful Life estimation. In *IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 88–95, 2017.

[11] J. Caceres, D. Gonzalez, T. Zhou, and E. L. Droguett. A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties. *Structural Control and Health Monitoring*, 28(10):e2811, 2021.

[12] N. DeVol, C. Saldana, and K. Fu. Inception Based Deep Convolutional Neural Network for Remaining Useful Life Estimation of Turbofan Engines. *Annual Conference of the PHM Society*, 13(1), 2021.

[13] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[14] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*, 2017.

[15] A. Boujamza and S. L. Elhaq. Attention-based LSTM for Remaining Useful Life Estimation of Aircraft Engines. *IFAC-PapersOnLine*, 55(12):450–455, 2022.

[16] S. Xiang, Y. Qin, J. Luo, H. Pu, and B. Tang. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliability Engineering & System Safety*, 216:107927, 2021.

[17] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[18] Z. Xie, S. Du, J. Lv, Y. Deng, and S. Jia. A Hybrid Prognostics Deep Learning Model for Remaining Useful Life Prediction. *Electronics*, 10(1):39, 2021.

[19] A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management*, pages 1–9, 2008.

[20] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183:240–251, 2019.

[21] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015:793161, 2015.

[22] M. Kim and K. Liu. A Bayesian Deep Learning Framework for Interval Estimation of Remaining Useful Life in Complex Systems by Incorporating General Degradation Characteristics. *IISE Transactions*, pages 1–23, 2020.